

# Integrated Algorithm for Decreasing Active Power Loss

K. Lenin

Department of EEE Prasad V. Potluri Siddhartha Institute of Technology,  
Kanuru, Vijayawada, Andhra Pradesh, India

---

## Article Info

### Article history:

Received Nov 22, 2017

Revised Jan 20, 2018

Accepted Feb 23, 2018

---

### Keyword:

Cuckoo Search

Firefly Algorithm

Integrated Algorithm

Optimal reactive power

Particle Swarm Optimization

Transmission loss

---

## ABSTRACT

This paper projects an Integrated Algorithm (IA) for solving optimal reactive power problem. Quick convergence of the Cuckoo Search (CS), the vibrant root change of the Firefly Algorithm (FA), and the incessant position modernization of the Particle Swarm Optimization (PSO) has been combined to form the Integrated Algorithm (IA). In order to evaluate the efficiency of the proposed Integrated Algorithm (IA), it has been tested in standard IEEE 57,118 bus systems and compared to other standard reported algorithms. Simulation results show that Integrated Algorithm (IA) is considerably reduced the real power loss and voltage profile within the limits.

Copyright © 2018 Institute of Advanced Engineering and Science.  
All rights reserved.

---

### Corresponding Author:

K. Lenin

Department of EEE Prasad V. Potluri Siddhartha Institute of Technology,  
Kanuru, Vijayawada, Andhra Pradesh, India.

Email: gklenin@gmail.com

---

## 1. INTRODUCTION

Optimal reactive power dispatch problem is one of the complex optimization problems in power system. The reactive power dispatch problem engross the best utilization of the existing generator bus voltage magnitudes, transformer tap setting and the output of reactive power sources to minimize the loss and to enhance the voltage stability of the system. Various mathematical techniques have been adopted to solve this optimal reactive power dispatch problem. These include the gradient method [1, 2], Newton method [3] and linear programming [4-7]. The gradient and Newton methods suffer from the difficulty in handling inequality constraints. To apply linear programming, the input- output function is to be expressed as a set of linear functions which may lead to loss of accuracy. Recently Global Optimization techniques such as genetic algorithms have been proposed to solve the reactive power flow problem [8, 9]. In recent years, the problem of voltage stability and voltage collapse has become a major concern in power system planning and operation. To enhance the voltage stability, voltage magnitudes alone will not be a reliable indicator of how far an operating point is from the collapse point [10]. The reactive power support and voltage problems are intrinsically related. Voltage stability evaluation using modal analysis [10] is used as the indicator of voltage stability. Recently growing popularity of the hybridization of different algorithmic concepts has been to obtain better performing systems that develop and combine the advantages of the individual pure strategies, that is, hybrids are believed to benefit from synergy. In fact, choosing an adequate combination of multiple algorithmic concepts is often the key to achieving top performance in solving many hard optimization problems [11-28]. This paper projects an Integrated Algorithm (IA) for solving optimal reactive power problem. Quick convergence of the Cuckoo Search (CS), the vibrant root change of the Firefly Algorithm (FA), and the incessant position modernization of the Particle Swarm Optimization (PSO) has been combined to form the Integrated Algorithm (IA). The aim of the paper is to improve the search features of three different metaheuristic algorithms, cuckoo search (CS), firefly algorithm (FA) and Particle Swarm Optimization (PSO). Where the cuckoo birds experience new places (arbitrary walk) utilizing firefly algorithm strategy instead of

Lévy flight. In this algorithm cuckoo birds will also be aware of each other positions utilizing PSO swarm communication technique to search for a better solution. In order to evaluate the efficiency of the proposed Integrated Algorithm (IA), it has been tested in standard IEEE 57,118 bus systems and compared to other standard reported algorithms. Simulation results show that Integrated Algorithm (IA) is considerably reduced the real power loss and voltage profile within the limits.

## 2. OBJECTIVE FUNCTION

Active power loss

The objective of the reactive power dispatch problem is to minimize the active power loss and can be defined in equations as follows:

$$F = P_L = \sum_{k \in \text{Nbr}} g_k (V_i^2 + V_j^2 - 2V_i V_j \cos \theta_{ij}) \quad (1)$$

Where F- objective function, PL – power loss,  $g_k$  - conductance of branch,  $V_i$  and  $V_j$  are voltages at buses  $i, j$ , Nbr- total number of transmission lines in power systems.

Voltage profile improvement

To minimize the voltage deviation in PQ buses, the objective function (F) can be written as:

$$F = P_L + \omega_v \times VD \quad (2)$$

Where VD - voltage deviation,  $\omega_v$ - is a weighting factor of voltage deviation.

And the Voltage deviation given by:

$$VD = \sum_{i=1}^{N_{pq}} |V_i - 1| \quad (3)$$

Where  $N_{pq}$ - number of load buses

Equality Constraint

The equality constraint of the problem is indicated by the power balance equation as follows:

$$P_G = P_D + P_L \quad (4)$$

Where  $P_G$ - total power generation,  $P_D$  - total power demand.

Inequality Constraints

The inequality constraint implies the limits on components in the power system in addition to the limits created to make sure system security. Upper and lower bounds on the active power of slack bus ( $P_g$ ), and reactive power of generators ( $Q_g$ ) are written as follows:

$$P_{g\text{slack}}^{\min} \leq P_{g\text{slack}} \leq P_{g\text{slack}}^{\max} \quad (5)$$

$$Q_{gi}^{\min} \leq Q_{gi} \leq Q_{gi}^{\max}, i \in N_g \quad (6)$$

Upper and lower bounds on the bus voltage magnitudes ( $V_i$ ) are given by:

$$V_i^{\min} \leq V_i \leq V_i^{\max}, i \in N \quad (7)$$

Upper and lower bounds on the transformers tap ratios ( $T_i$ ) are given by:

$$T_i^{\min} \leq T_i \leq T_i^{\max}, i \in N_T \quad (8)$$

Upper and lower bounds on the compensators ( $Q_c$ ) are given by:

$$Q_c^{\min} \leq Q_c \leq Q_c^{\max}, i \in N_c \quad (9)$$

Where  $N$  is the total number of buses,  $N_g$  is the total number of generators,  $N_T$  is the total number of Transformers,  $N_c$  is the total number of shunt reactive compensators.

### 3. CUCKOO SEARCH ALGORITHM (CS)

The Cuckoo Search Algorithm (CS) was inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of host birds. Some cuckoos have evolved in such a way that female parasitic cuckoos can imitate the colours and patterns of the eggs of a few chosen host species. This reduces the probability of the eggs being abandoned and, therefore, increases their re-productivity. In general, the cuckoo eggs hatch slightly earlier than their host eggs. Once the first cuckoo chick is hatched, his first instinct action is to evict the host eggs by blindly propelling the eggs out of the nest. This action results in increasing the cuckoo chick's share of food provided by its host bird. Moreover, studies show that a cuckoo chick can imitate the call of host chicks to gain access to more feeding opportunity. The CS models such breeding behavior and, thus, can be applied to various optimization problems.

#### 3.1. Levy Flights

In nature, animals search for food in a random or quasi random manner. Generally, the foraging path of an animal is effectively a random walk because the next move is based on both the current location/state and the transition probability to the next location. The chosen direction implicitly depends on a probability, which can be modeled mathematically. Various studies have shown that the flight behavior of many animals and insects demonstrates the typical characteristics of Lévy flights. A Lévy flight is a random walk in which the step-lengths are distributed according to a heavy-tailed probability distribution. After a large number of steps, the distance from the origin of the random walk tends to a stable distribution.

#### 3.2. Cuckoo Search Implementation

Each egg in a nest represents a solution, and a cuckoo egg represents a new solution. The aim is to employ the new and potentially better solutions (cuckoos) to replace not-so-good solutions in the nests. In the simplest form, each nest has one egg. The algorithm can be extended to more complicated cases in which each nest has multiple eggs representing a set of solutions. The CS is based on three idealized rules:

1. Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest;
2. The best nests with high quality of eggs (solutions) will carry over to the next generations;
3. The number of available host nests is fixed, and a host can discover an alien egg with probability  $p_a \in [0,1]$ . In this case, the host bird can either throw the egg away or abandon the nest to build a completely new nest in a new location.

For simplicity, the last assumption can be approximated by a fraction  $p_a$  of the  $n$  nests being replaced by new nests, having new random solutions. For a maximization problem, the quality or fitness of a solution can simply be proportional to the objective function. Other forms of fitness can be defined in a similar way to the fitness function in genetic algorithms.

Based on the above-mentioned rules, the basic steps of the CS can be summarized as code below:

```

Begin
Objective function  $f(x)$ ,  $x = (x_1, \dots, x_d)^T$ 
Generate initial population of  $n$  host nests  $x_i$ , ( $i = 1, 2, \dots, n$ )
While (t Max Generation) or (stop criterion)
  Get a cuckoo randomly by Levy flights, Evaluate its quality / fitness  $F_i$ 
  Choose a nest among  $n$  (say  $j$ ) arbitrarily, If ( $F_i > F_j$ ) replace  $j$  by the new solution;
End if
A fraction ( $p_a$ ) of worse nests are abandoned and new ones are built;
Keep the best solutions (or nests with quality solutions);
Rank the solutions and find the current best
End while
Post process results and visualization
End

```

When generating new solutions  $x_i(t+1)$  for the  $i$ th cuckoo, the following Lévy flight is performed by,

$$x_i(t+1) = x_i(t) + \alpha \odot \text{Levy}(t^{-\lambda}, 1 < \lambda \leq 3\lambda) \quad (10)$$

where  $\alpha > 0$  is the step size, which should be related to the scale of the problem of interest. The product  $\odot$  means entry-wise multiplications. In this research work, we consider a Lévy flight in which the step-lengths are distributed according to the following probability distribution

$$\text{Levy } u = t^{-\lambda}, 1 < \lambda \leq 3 \quad (11)$$

which has an infinite variance. Here, the consecutive jumps/steps of a cuckoo essentially form a random walk process which obeys a power-law step-length distribution with a heavy tail. It is worth pointing out that, in the real world, if a cuckoo's egg is very similar to a host's eggs, then this cuckoo's egg is less likely to be discovered, thus the fitness should be related to the difference in solutions. Therefore, it is a good idea to do a random walk in a biased way with some random step sizes.

#### 4. FIREFLY ALGORITHM (FA)

In this method, each solution in a population represents a solution which is located randomly within a specified searching space. The  $i$ th solution,  $X_i$ , is represented as follows:

$$X_{i(t)} = \{X_{i1(t)}, X_{i2(t)}, \dots, X_{id(t)}\} \quad (12)$$

Where  $X_{i(t)}$  is the vector with  $k = 1, 2, 3, \dots, d$ , and  $t$  is the time step. Initially, the fitness value of each solution was evaluated. The solution that produced the best fitness value would be chosen as the current best solution in the population. Then, a sorting operation was performed. In this operation, the newly evaluated solutions were ranked based on the fitness values and divided into two sub-populations. The first sub-population contained solutions that produced potential fitness values. The fitness value of each  $i$ th solution in this sub-population was then compared with its  $j$ th neighbouring solution. If the fitness value of the neighbouring solution was better, the distance between every solution would then be calculated using the standard Euclidean distance measure. The distance was used to compute the attractiveness,  $\beta$ :

$$\beta = \beta_0 e^{-\gamma r_{ij}^2} \quad (13)$$

Where  $\beta_0$ ,  $\gamma$  and  $r_{ij}$  are the predefined attractiveness, light absorption coefficient, and distance between  $i$ th solution and its  $j$ th neighbouring solution, respectively. Later, this new attractiveness value was used to update the position of the solution, as follows:

$$x_{id} = x_{id} + \beta(x_{jd} - x_{id}) + \alpha\left(\delta - \frac{1}{2}\right) \quad (14)$$

Where  $\alpha$  and  $\delta$  are uniformly distributed random values between 0 to 1. Thus, the updated attractiveness values assisted the population to move towards the solution that produced the current best fitness value.

On the other hand, the second sub-population contained solutions that produced less significant fitness values. The solutions in this population were subjected to undergo the evolutionary operations of Differential Evolution method. Firstly, the trivial solutions were produced by the mutation operation performed on the original counterparts. The  $i$ th trivial solution,  $V_i$ , was generated based on the following equation:

$$V_{i(t)} = \{v_{i1(t)}, v_{i2(t)}, \dots, v_{id(t)}\} \quad (15)$$

$$v_{i(t)} = x_{best(t)} + F \cdot (x_{r1(t)} - x_{r2(t)}) \quad (16)$$

Where  $x_{best(t)}$  is the vector of current best solution,  $F$  is the mutation factor,  $x_{r1(t)}$  and  $x_{r2(t)}$  are randomly chosen vectors from the neighbouring solutions. Next, the offspring solution was produced by the crossover operation that involved the parent and the trivial solution. The vectors of the  $i$ th offspring solution,  $Y_i$ , were created as follows

$$Y_{i(t)} = \{y_{i1(t)}, y_{i2(t)}, \dots, y_{id(t)}\} \quad (17)$$

$$y_{i(t)} = \begin{cases} v_{i(t)} & \text{if } R < CR \\ x_{i(t)} & \text{otherwise} \end{cases} \quad (18)$$

Where  $R$  is a uniformly distributed random value between 0 to 1 and  $CR$  is the predefined crossover constant. As the population of the offspring solution was produced, a selection operation was required to keep the population size constant. The operation was performed as follows:

$$X_{i(t+1)} = \begin{cases} Y_{i(t)} & \text{if } f(Y_{i(t)}) \leq f(X_{i(t)}) \\ X_{i(t)} & \text{if } f(Y_{i(t)}) > f(X_{i(t)}) \end{cases} \quad (19)$$

This indicates that the original solution would be replaced by the offspring solution if the fitness value of the offspring solution was better than the original solution. Otherwise, the original solution would remain in the population for the next iteration. The whole procedure was repeated until the stopping criterion was met.

#### Firefly Algorithm

*Input: Randomly initialized position of d dimension problem:  $X_i$*

*Output: Position of the approximate global optima:  $X_G$*

*Begin*

*Initialize population; Evaluate fitness value;*

*$X_G \leftarrow$  Select current best solution;*

*For  $t \leftarrow 1$  to max*

*Sort population based on the fitness value;*

*$X_{good} \leftarrow first_{half}(X); X_{worst} \leftarrow second\_half(X);$*

*For  $i \leftarrow 0$  to number of  $X_{good}$  solutions*

*For  $j \leftarrow 0$  to number of  $X_{good}$  solutions*

*If  $(f(X_i) > f(X_j))$  then*

*Calculate distance and attractiveness;*

*Update position;*

*End If*

*End For*

*End For*

*For  $i \leftarrow 0$  to number of  $X_{worst}$  solutions*

*Create trivial solution,  $V_{i(t)}$ ;*

*Perform crossover,  $Y_{i(t)}$ ;*

*Perform selection,  $X_{i(t)}$ ;*

*End For*

*$X \leftarrow combine(X_{good}, X_{worst});$*

*$X_G \leftarrow$  Select current best solution;*

*$t \leftarrow t + 1$  ; 1;*

*End For*

*End Begin*

## 5. PARTICLE SWARM OPTIMIZATION (PSO)

Particle Swarm Optimization is a population-based optimization algorithm inspired by the behaviour of flocks of birds. It was firstly introduced by Kennedy and Eberhart and it has been largely applied to solve optimization problems. The standard approach is composed by a swarm of particles, where each one has a position within the search space  $\vec{x}_i$  and each position represents a solution for the problem. The particles fly through the search space of the problem searching for the best solution, according to the current velocity  $\vec{v}_i$  the best position found by the particle itself ( $\vec{p}_{besti}$ ) and the best position found by the entire swarm during the search so far ( $\vec{G}_{besti}$ ). According to the approach proposed by Shi and Eberhart (this approach is also called inertia PSO), the velocity of a particle is evaluated at each iteration of the algorithm by using the following equation:

$$\vec{v}_i(t+1) = \omega \vec{v}_i(t) + r_1 c_1 |\vec{p}_{besti} - \vec{x}_i(t)| + r_2 c_2 |\vec{G}_{besti} - \vec{x}_i(t)| \quad (20)$$

Where  $r_1$  and  $r_2$  are numbers randomly generated in the interval  $[0, 1]$ . The inertia weight ( $\omega$ ) controls the influence of the previous velocity and balances the exploration-exploitation behaviour along the process. It generally decreases from 0.9 to 0.4 during the algorithm execution.  $c_1$  &  $c_2$  are called cognitive and social acceleration constants, respectively, and weights the influence of the memory of the particle and the information acquired from the neighbourhood. The position of each particle is updated based on the velocity of the particle, according to the following equation:

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \quad (21)$$

The communication topology defines the neighbourhood of the particles and, as a consequence, the flow of information through the particles. There are two basic topologies: global and local. In the former, each particle shares and acquires information directly from all other particles, i.e. all particles use the same social memory, called  $G_{best}$ . In the local topology, each particle only shares information with two neighbours and the social memory is not the same within the whole swarm. This approach, called  $L_{best}$ , helps to avoid a premature attraction of all particles to a single spot point in the search space.

### 5.1. Integrated Algorithm (IA) for solving optimal reactive power problem

In this proposed Integrated Algorithm (IA), cuckoo bird will be able to perform stochastic behaviour (random walk) using the strategy of firefly algorithm, instead of using Lévy Flight movement. Also the cuckoo birds will be able to communicate them to inform each other from their position and help each other to immigrate to a better place. Each cuckoo bird will record the best personal experience as pbest during its own life. In addition, the best pbest among all the birds is called gbest. The cuckoo birds' communication is established through the pbest and gbest. They update their position using these parameters along with the velocity of each swarm member. The update rule for cuckoo (i's) position is carried out according to equations (20, 21).

1. Start
2. Initiate a random population of  $n$  host
3. Get a cuckoo randomly  $i$
4. Evaluation its fitness,  $F_i$
5. Select a nest among  $n$  randomly,  $j$
6.  $F_i < F_j$  ; if yes , Replace  $j$  by the new solution  
If No, Let  $j$  as the solution
7. Move cuckoo birds using equation (20,21) in PSO
8. Abandon a fraction,  $Pa$  of worse nests and build new ones at new locations by using (12-19) firefly.
9. Keep the current best
10.  $t \leq \maxIterations$  ; if No go to step 3  
if Yes keep the current best
11. End

## 6. SIMULATION RESULTS

At first Integrated Algorithm (IA) has been tested in standard IEEE-57 bus power system. The reactive power compensation buses are 18, 25 and 53. Bus 2, 3, 6, 8, 9 and 12 are PV buses and bus 1 is selected as slack-bus. The system variable limits are given in Table 1.

The preliminary conditions for the IEEE-57 bus power system are given as follows:

$P_{load} = 12.104$  p.u.  $Q_{load} = 3.040$  p.u.

The total initial generations and power losses are obtained as follows:

$\sum P_G = 12.418$  p.u.  $\sum Q_G = 3.3122$  p.u.

$P_{loss} = 0.25801$  p.u.  $Q_{loss} = -1.2008$  p.u.

Table 2 shows the various system control variables i.e. generator bus voltages, shunt capacitances and transformer tap settings obtained after optimization which are within the acceptable limits. In Table 3, shows the comparison of optimum results obtained from proposed methods with other optimization techniques. These results indicate the robustness of proposed approaches for providing better optimal solution in case of IEEE-57 bus system.

Table 1. Variable Limits

Reactive Power Generation Limits							
Bus no	1	2	3	6	8	9	12
Qgmin	-1.4	-.015	-.02	-0.04	-1.3	-0.03	-0.4
Qgmax	1	0.3	0.4	0.21	1	0.04	1.50
Voltage And Tap Setting Limits							
vgmin	Vgmax	vpqmin	Vpqmax	tkmin	tkmax		
0.9	1.0	0.91	1.05	0.9	1.0		
Shunt Capacitor Limits							
Bus no	18			25		53	
Qcmin	0			0		0	
Qcmax	10			5.2		6.1	

Table 2. Control variables obtained after optimization

Control Variables	IA
V1	1.1
V2	1.031
V3	1.037
V6	1.028
V8	1.024
V9	1.000
V12	1.012
Qc18	0.0661
Qc25	0.201
Qc53	0.0472
T4-18	1.000
T21-20	1.042
T24-25	0.861
T24-26	0.871
T7-29	1.051
T34-32	0.871
T11-41	1.010
T15-45	1.031
T14-46	0.910
T10-51	1.020
T13-49	1.060
T11-43	0.910
T40-56	0.900
T39-57	0.950
T9-55	0.950

Table 3. Comparison results

S.No.	Optimization Algorithm	Finest Solution	Poorest Solution	Normal Solution
1	NLP [29]	0.25902	0.30854	0.27858
2	CGA [29]	0.25244	0.27507	0.26293
3	AGA [29]	0.24564	0.26671	0.25127
4	PSO-w [29]	0.24270	0.26152	0.24725
5	PSO-cf [29]	0.24280	0.26032	0.24698
6	CLPSO [29]	0.24515	0.24780	0.24673
7	SPSO-07 [29]	0.24430	0.25457	0.24752
8	L-DE [29]	0.27812	0.41909	0.33177
9	L-SACP-DE [29]	0.27915	0.36978	0.31032
10	L-SaDE [29]	0.24267	0.24391	0.24311
11	SOA [29]	0.24265	0.24280	0.24270
12	LM [30]	0.2484	0.2922	0.2641
13	MBEP1 [30]	0.2474	0.2848	0.2643
14	MBEP2 [30]	0.2482	0.283	0.2592
15	BES100 [30]	0.2438	0.263	0.2541
16	BES200 [30]	0.3417	0.2486	0.2443
17	Proposed IA	0.22016	0.23028	0.22208

Then Integrated Algorithm (IA) has been tested in standard IEEE 118-bus test system [31]. The system has 54 generator buses, 64 load buses, 186 branches and 9 of them are with the tap setting transformers. The limits of voltage on generator buses are 0.95 -1.1 per-unit., and on load buses are 0.95 -1.05 per-unit. The limit of transformer rate is 0.9 -1.1, with the changes step of 0.025. The limitations of reactive power source are listed in Table 4, with the change in step of 0.01.

Table 4. Limitation of reactive power sources

BUS	5	34	37	44	45	46	48
QCMAX	0	14	0	10	10	10	15
QCMIN	-40	0	-25	0	0	0	0
BUS	74	79	82	83	105	107	110
QCMAX	12	20	20	10	20	6	6
QCMIN	0	0	0	0	0	0	0

The statistical comparison results of 50 trial runs have been list in Table 5 and the results clearly show the better performance of proposed Integrated Algorithm (IA) in reducing the real power loss.

Table 5. Comparison results

Active power loss (MW)	BBO [32]	ILSBBO/strategy1 [32]	ILSBBO/strategy1[32]	Proposed IA
Min	128.77	126.98	124.78	114.38
Max	132.64	137.34	132.39	118.62
Average	130.21	130.37	129.22	116.58

## 7. CONCLUSION

In this research paper Integrated Algorithm (IA) is successfully solved optimal reactive power problem. Quick convergence of the Cuckoo Search (CS), the vibrant root change of the Firefly Algorithm (FA), and the incessant position modernization of the Particle Swarm Optimization (PSO) has been combined to form the Integrated Algorithm (IA). In order to evaluate the efficiency of the proposed Integrated Algorithm (IA), it has been tested in standard IEEE 57,118 bus systems and compared to other standard reported algorithms. Simulation results show that Integrated Algorithm (IA) is considerably reduced the real power loss and voltage profile within the limits.

## REFERENCES

- [1] O.Alsac, and B. Scott, "Optimal load flow with steady state security", *IEEE Transaction*. PAS -1973, pp. 745-751.
- [2] Lee K Y, Paru Y M, Ortiz J L –A united approach to optimal real and reactive power dispatch, *IEEE Transactions on power Apparatus and systems* 1985: PAS-104: 1147-1153
- [3] A.Monticelli, M .V.F Pereira, and S. Granville, "Security constrained optimal power flow with post contingency corrective rescheduling", *IEEE Transactions on Power Systems: PWRS-2*, No. 1, pp.175-182.,1987.
- [4] DeebN, Shahidehpur S.M, Linear reactive power optimization in a large power network using the decomposition approach. *IEEE Transactions on power system* 1990: 5(2): 428-435
- [5] Hobson, "Network constrained reactive power control using linear programming," *IEEE Transactions on power systems* PAS -99 (4) ,pp 868=877, 1980
- [6] K.Y Lee, Y.M Park, and J.L Ortiz, "Fuel –cost optimization for both real and reactive power dispatches", *IEE Proc*; 131C, (3), pp.85-93.
- [7] M.K. Mangoli, and K.Y. Lee, "Optimal real and reactive power control using linear programming", *Electr.Power Syst.Res*, Vol.26, pp.1-10,1993.
- [8] K.Anburaja, "Optimal power flow using refined genetic algorithm", *Electr.Power Compon.Syst*, Vol. 30, 1055-1063, 2002.
- [9] Devaraj, and B. Yeganarayana, "Genetic algorithm based optimal power flow for security enhancement", *IEE proc-Generation.Transmission and. Distribution*; 152, 6 November 2005.
- [10] C.A. Canizares, A.C.Z.de Souza and V.H. Quintana, "Comparison of performance indices for detection of proximity to voltage collapse," vol. 11. no.3, pp.1441-1450, Aug 1996.
- [11] Eberhart, R.C. and J. Kennedy, 1995. *A new optimizer using particle swarm theory*. Proceeding of the 6th International Symposium on Micro Machine and Human Science, pp: 39-43.
- [12] Abdullah, A. et al. 2012. A new hybrid firefly algorithm for complex and nonlinear problem. *Distributed Computing and Artificial Intelligence*. Springer. 673–680.
- [13] Bratton, D. and Kennedy, J. 2007. *Defining a standard for particle swarm optimization*. Swarm Intelligence Symposium, 2007. SIS 2007. IEEE (2007), 120–127.
- [14] El-Sawy, A.A. et al. 2012.A Novel Hybrid Ant Colony Optimization and Firefly Algorithm for Solving Constrained Engineering Design Problems. *Journal of Natural Sciences and Mathematics*.6, (2012).
- [15] Farook, S. and Raju, P.S. Evolutionary Hybrid Genetic-Firefly Algorithm for Global Optimization.
- [16] Gandomi, A.H. et al. 2013. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with Computers*. 29, (2013), 17–35.
- [17] Ghodrati, A. and Lotfi, S. 2012. A hybrid CS/PSO algorithm for global optimization.*Intelligent Information and Database Systems*. Springer.89–98.
- [18] Hassanzadeh, T. and Meybodi, M.R. 2012. *A new hybrid algorithm based on Firefly Algorithm and cellular learning automata*. 20th Iranian Conference on Electrical Engineering (ICEE) (2012), 628–633.
- [19] Hezam, I.M. and AbdelRaouf, O. 2013. Particle Swarm Optimization Approach for Solving Complex Variable Fractional Programming Problems. *International Journal of Engineering*. 2, (2013).
- [20] Hezam, I.M. and AbdelRaouf, O.M.M.H. 2013. Solving Fractional Programming Problems Using Metaheuristic Algorithms under Uncertainty. *International Journal of Advanced Computing*. 46, (2013), 1261–1270.
- [21] Hezam, I.M. and Raouf, O.A. 2013.Employing Three Swarm Intelligent Algorithms for Solving Integer Fractional Programming Problems. *International Journal of Scientific and Engineering Research (IJSER)*. 4, Issue 7, (2013), 191–198.
- [22] Jones, K.O. and Boizanté, G. 2011. *Comparison of Firefly algorithm optimisation, particle swarm optimisation and differential evolution*. Proceedings of the 12th International Conference on Computer Systems and Technologies (2011), 191–197.
- [23] Lee, K.Y. and El-Sharkawi, M.A. 2008. Modern heuristic optimization techniques: theory and applications to power systems. Wiley-IEEE press.



- [24] Raju, R. et al. 2013. Hybrid Ant Colony Optimization and Cuckoo Search Algorithm for Job Scheduling. *Advances in Computing and Information Technology. Springer*. 491–501.
- [25] Wang, F. et al. 2011. *Hybrid optimization algorithm of PSO and Cuckoo Search. Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*, 2011 2nd International Conference on (2011), 1172–1175.
- [26] Wang, G. et al. 2012. A hybrid meta-heuristic DE/CS algorithm for UCAV path planning. *Journal of Information and Computational Science*. 5, (2012), 4811–4818.
- [27] Yang, X.-S. 2010. *Engineering optimization: an introduction with metaheuristic applications*. Wiley.
- [28] Yang, X.-S. 2011. *Nature-inspired metaheuristic algorithms*. Luniver Press.
- [29] Chaohua Dai, Weirong Chen, Yunfang Zhu, and Xuexia Zhang, “Seeker optimization algorithm for optimal reactive power dispatch,” *IEEE Trans. Power Systems*, Vol. 24, No. 3, August 2009, pp. 1218-1231.
- [30] J. R. Gomes and O. R. Saavedra, “*Optimal reactive power dispatch using evolutionary computation: Extended algorithms*,” *IEE Proc.-Gener. Transm. Distrib.* Vol. 146, No. 6. Nov. 1999.
- [31] IEEE, “The IEEE 30-bus test system and the IEEE 118-test system”, (1993), <http://www.ee.washington.edu/trsearch/pstca/>.
- [32] Jiangtao Cao, Fuli Wang and Ping Li, “An Improved Biogeography-based Optimization Algorithm for Optimal Reactive Power Flow” *International Journal of Control and Automation* Vol.7, No.3 (2014), pp.161-176.